A3_ComponentesPrincipales

Josue Salvador Cano Martinez

2022-10-11

```
# Matriz de datos
x1 = c(2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1)
x2 = c(2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9)
M <- data.frame(x1, x2)
print (M)
```

```
##     x1  x2
## 1  2.5 2.4
## 2  0.5 0.7
## 3  2.2 2.9
## 4  1.9 2.2
## 5  3.1 3.0
## 6  2.3 2.7
## 7  2.0 1.6
## 8  1.0 1.1
## 9  1.5 1.6
## 10 1.1 0.9
```

```
# Creación de matriz con dos columnas que contiene los datos de los registros
```

```
# Matriz de datos centrados en sus medias
meanX1 = c(rep(mean(x1), 10))
meanX2 = c(rep(mean(x2), 10))

M1 <- data.frame(meanX1, meanX2)
M1 = M - M1
print(M1)
```

```
##       x1    x2
## 1   0.69  0.49
## 2  -1.31 -1.21
## 3   0.39  0.99
## 4   0.09  0.29
## 5   1.29  1.09
## 6   0.49  0.79
## 7   0.19 -0.31
## 8  -0.81 -0.81
## 9  -0.31 -0.31
## 10 -0.71 -1.01
```

```
# Se resta a cada valor la media de la variable a la que pertenece. Con esto se consigue central
izar las variables y que su media sea 0
```

```
# Matriz de varianza-covarianza de la matriz de datos centrados
mcov = cov(M1)
print(mcov)
```

```
##           x1        x2
## x1 0.6165556 0.6154444
## x2 0.6154444 0.7165556
```

```
# Se calcula la matriz de correlaciones entre cada par de variables. Debido a que hay dos variab
les el resultado es una matriz simétrica de 2x2
```

```
# Valores propios y vectores propios de la matriz de varianza-covarianza de la matriz de datos c
entrados
val = eigen(mcov)$values
print(val)
```

```
## [1] 1.2840277 0.0490834
```

```
vec =eigen(mcov)$vectors
print(vec)
```

```
##           [,1]       [,2]
## [1,] 0.6778734 -0.7351787
## [2,] 0.7351787  0.6778734
```

```
# Debido a que la matriz de covarianzas es cuadrada, es posible calcular sus correspondientes ei
genvectors y eigenvalues
```

```
# Matriz transpuesta de los vectores propios y matriz traspuesta de datos centrados
t_v = t(vec)
print(t_v)
```

```
##            [,1]      [,2]
## [1,]  0.6778734 0.7351787
## [2,] -0.7351787 0.6778734
```

```
t_M1 = t(M1)
print(t_M1)
```

```
##    [,1]  [,2] [,3] [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10]
## x1 0.69 -1.31 0.39 0.09 1.29 0.49  0.19 -0.81 -0.31 -0.71
## x2 0.49 -1.21 0.99 0.29 1.09 0.79 -0.31 -0.81 -0.31 -1.01
```

```
# Producto de la matriz transpuesta de los vectores propios con la transpuesta de la matriz de d
atos centrados
CP = t_v%*%t_M1
rownames(CP)= c("CP1", "CP2")
t(CP)
```

```
##               CP1         CP2
##  [1,]  0.82797019 -0.17511531
##  [2,] -1.77758033  0.14285723
##  [3,]  0.99219749  0.38437499
##  [4,]  0.27421042  0.13041721
##  [5,]  1.67580142 -0.20949846
##  [6,]  0.91294910  0.17528244
##  [7,] -0.09910944 -0.34982470
##  [8,] -1.14457216  0.04641726
##  [9,] -0.43804614  0.01776463
## [10,] -1.22382056 -0.16267529
```

```
# El proceso en que se multiplicaron los eigenvectors transpuestos por los datos originales cent
rados y también transpuestos dio lugar a obtener los componentes principales. Lo que se observa
 abajo es el valor que toma cada componente para cada observación en función de las variables or
iginales (principal component scores)
```

```
cpa <- prcomp(M, scale=TRUE)
names(cpa)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
print(cpa)
```

```
## Standard deviations (1, .., p=2):
## [1] 1.3877785 0.2721594
##
## Rotation (n x k) = (2 x 2):
##           PC1        PC2
## x1 -0.7071068  0.7071068
## x2 -0.7071068 -0.7071068
```

```
# Reducción de la dimensionalidad de las variables, se tienen dos, por lo que se puede deducir q
ue hay un componente que explica la varianza de las dos variables.
```

```
print("desviaciones estándar: ")
```

```
## [1] "desviaciones estándar: "
```

```
cpa$sdev
```

```
## [1] 1.3877785 0.2721594
```

```
print("medias: ")
```

```
## [1] "medias: "
```

```
print("center y scale dan las medias y desv estándar previa estandarización: ")
```

```
## [1] "center y scale dan las medias y desv estándar previa estandarización: "
```

```
cpa$center
```

```
##   x1   x2
## 1.81 1.91
```

```
cpa$scale
```

```
##        x1        x2
## 0.7852105 0.8464960
```

```
print("Los coeficientes de la combinación lineal normalizada de componete")
```

```
## [1] "Los coeficientes de la combinación lineal normalizada de componete"
```

```
cpa$rotation
```

```
##           PC1        PC2
## x1 -0.7071068  0.7071068
## x2 -0.7071068 -0.7071068
```

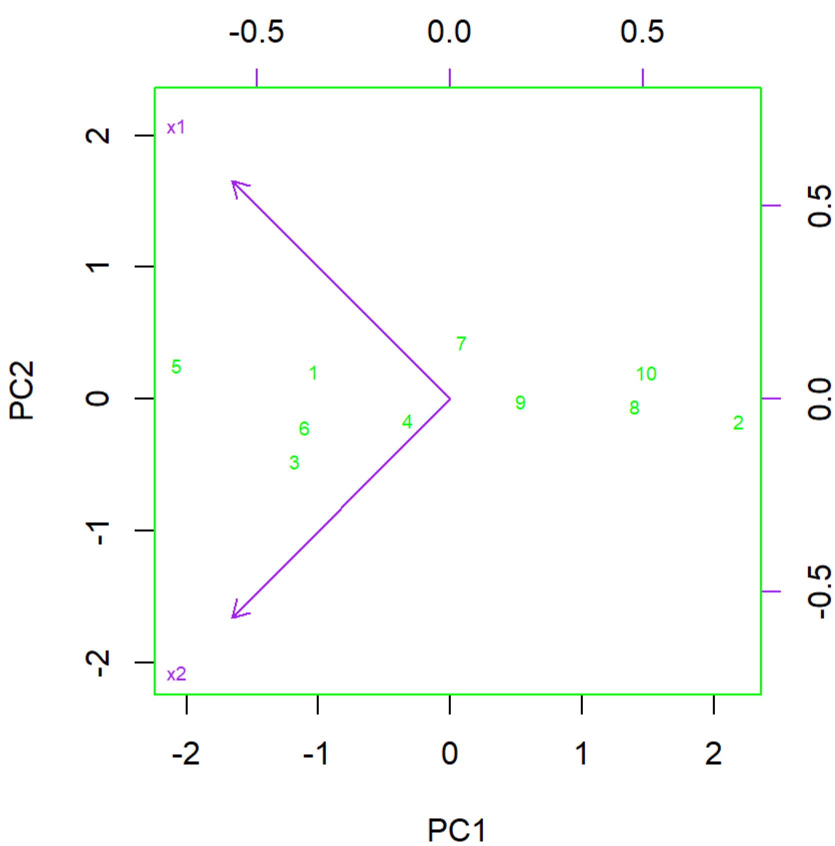
```
print("Los datos por sustituidos en la combinación lineal de vectores propios:")
```

```
## [1] "Los datos por sustituidos en la combinación lineal de vectores propios:"
```

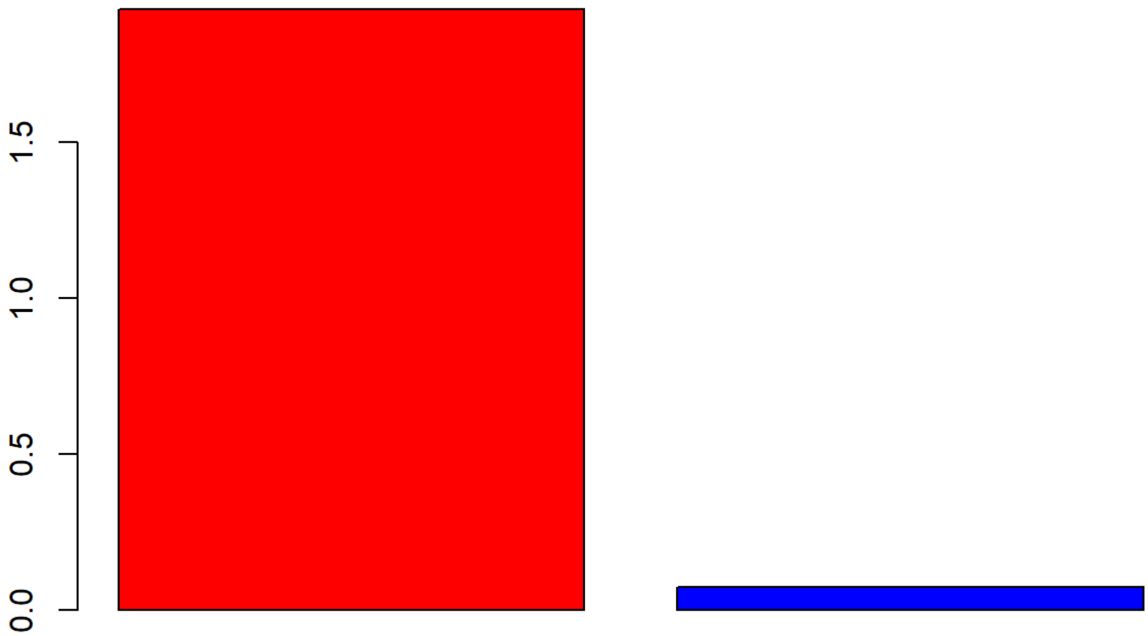
```
cpa$x
```

```
##               PC1         PC2
##  [1,] -1.03068029  0.21205314
##  [2,]  2.19045016 -0.16894230
##  [3,] -1.17818776 -0.47577321
##  [4,] -0.32329464 -0.16119898
##  [5,] -2.07219947  0.25117173
##  [6,] -1.10117414 -0.21865330
##  [7,]  0.08785251  0.43005447
##  [8,]  1.40605089 -0.05281009
##  [9,]  0.53811824 -0.02021127
## [10,]  1.48306451  0.20430982
```

```
biplot(x = cpa, scale = 0, cex = 0.6, col = c("green", "purple"))
```




```
barplot(cpa$sdev^2, col = c("red", "blue"))
```




```
# La gráfica de dispersión permite apreciar hacia qué componente pertenece cada registro
```

```
summary(cpa)
```

```
## Importance of components:
##                          PC1     PC2
## Standard deviation     1.388 0.27216
## Proportion of Variance 0.963 0.03704
## Cumulative Proportion  0.963 1.00000
```

```
# De acuerdo con la proporción de varianza explicada, el 96.3% de la varianza de los datos está
 integrada en el componente 1
```